Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:

receiving a source data schema and a target data schema, the source data schema being different from the target data schema;

mapping the source data schema into an ontology model;

mapping the target data schema into the ontology model; and

deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation ~~is adapted to~~ can transform the first data directly into the second data, and wherein the derived transformation ~~is adapted to~~ can transform the first data into the second data without transforming the first data via an ontological format.

2. (Original) The method of claim 1 further comprising converting at least one of the source data schema and the target schema from an external format to an internal format.

3. (Original) The method of claim 1 further comprising receiving the ontology model.

4. (Original) The method of claim 3 further comprising converting the ontology model from an external format to an internal format.

5. (Original) The method of claim 1 further comprising generating the ontology model.

6. (Original) The method of claim 5 further comprising receiving an initial ontology model, wherein said generating generates the ontology model from the initial ontology model.

7. (Original) The method of claim 6 further comprising converting the initial ontology model from an external format to an internal format.

8. (Previously Presented) The method of claim 1 further comprising automatically generating executable program code that transforms the first data into the second data.

9. (Original) The method of claim 1 wherein the source data schema is a source table schema describing source data tables, wherein the target data schema is a target table schema describing target data tables, and wherein the source table schema and the target table schema each describes at least one table having columns.

10. (Previously Presented) The method of claim 9 wherein the source table schema is a source relational database schema describing source relational database tables, wherein the target table schema is a target relational database schema describing target relational database tables, and wherein the automatically derived transformation is an SQL query.

11. (Original) The method of claim 10 wherein said mapping a source data schema and said mapping a target data schema each comprise:

identifying at least one class in the ontology model corresponding to at least one table; and

identifying at least one property or composition of properties in the ontology model corresponding to at least one table column.

12. (Previously Presented) The method of claim 11 wherein said deriving comprises:

labeling properties of the ontology model with symbols;

converting at least one column in the source relational database schema into at least one source symbol;

converting at least one column in the target relational database schema into at least one target symbol; and

expressing the at least one target symbol in terms of at least one source symbol.

13. (Original) The method of claim 12 wherein said expressing uses expressions involving composition of properties.

14. (Previously Presented) The method of claim 12 wherein at least one dependency exists among properties in the ontology model, and wherein said deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target relational

database columns and source relational database columns, and wherein said expressing incorporates the at least one dependency between target relational database columns and source relational database columns.

15. (Original) The method of claim 14 wherein said expressing uses expressions involving arithmetic operations.

16. (Original) The method of claim 14 wherein said expressing uses expressions involving character string operations.

17. (Original) The method of claim 10 further comprising applying the query to at least one source relational database table to populate at least one target relational database table.

18. (Original) The method of claim 17 wherein the at least one source relational database table reside in a single database.

19. (Original) The method of claim 17 wherein the at least one source relational database table reside in multiple databases.

20. (Original) The method of claim 1 wherein the source data schema is a source document schema describing source documents, and wherein the target data schema is a target document schema describing target documents.

21. (Original) The method of claim 20 wherein the source document schema is a source DTD describing source XML documents, wherein the target document schema is a target DTD describing target XML documents, and wherein the source DTD and the target DTD each describes at least one XML element or XML attribute.

22. (Original) The method of claim 21 wherein the transformation is an XQuery.

23. (Original) The method of claim 21 wherein the transformation is an XSLT script.

24. (Original) The method of claim 20 wherein the source document schema is a source XML schema describing source XML documents, wherein the target document schema is a target XML schema

describing target XML documents, and wherein the source XML schema and the target XML schema each describes at least one XML complexType having at least one XML element or XML attribute.

25. (Original) The method of claim 24 wherein the transformation is an XQuery.

26. (Original) The method of claim 24 wherein the transformation is an XSLT script.

27. (Original) The method of claim 24 wherein said mapping a source data schema and said mapping a target data schema each comprise:

identifying at least one class in the ontology model corresponding to at least one XML complexType; and

identifying at least one property or composition of properties in the ontology model corresponding to at least one XML element or XML attribute.

28. (Previously Presented) The method of claim 24 wherein said deriving comprises expressing XML elements and XML attributes of the target XML schema in terms of XML elements and XML attributes of the source XML schema.

29. (Original) The method of claim 28 wherein said expressing is performed recursively through XPath paths.

30. (Previously Presented) The method of claim 27 wherein at least one dependency exists among properties in the ontology model, and wherein said deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target XML elements and source XML elements.

31. (Original) The method of claim 26 further comprising applying the XSLT script to at least one source XML document to generate at least one target XML document.

32. (Original) The method of claim 31 wherein the at least one source XML document reside in a single database.

33. (Original) The method of claim 31 wherein the at least one source XML document reside in multiple databases.

34. (Currently Amended) A system for deriving transformations for transforming data from one data schema to another, comprising:

a schema receiver receiving a source data schema and a target data schema, the source data schema being different than the target data schema;

a mapping processor mapping a data schema into an ontology model; and

a transformation processor deriving a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, based on respective source mappings and target mappings generated by said mapping processor, wherein the source mappings map the source data schema into the ontology model and wherein the target mappings map the target data schema into the ontology model, wherein the derived transformation transforms the first data directly into the second data, and wherein the derived transformation ~~is adapted to~~ can transform the first data into the second data without transforming the first data via an ontological format.

35. (Previously Presented) The system of claim 34 further comprising a schema format converter converting at least one of the source data schema and the target data schema from an external format to an internal format.

36. (Original) The system of claim 34 further comprising an ontology receiver receiving the ontology model.

37. (Previously Presented) The system of claim 36 further comprising an ontology format converter, converting the ontology model from an external format to an internal format.

38. (Original) The system of claim 34 further comprising an ontology builder generating the ontology model.

39. (Original) The system of claim 38 further comprising an ontology receiver receiving an initial ontology model, wherein said ontology builder generates the ontology model from the initial ontology model.

40. (Previously Presented) The system of claim 39 further comprising an ontology format converter, converting the initial ontology model from an external format to an internal format.

41. (Previously Presented) The system of claim 34 further comprising a program code generator generating executable program code operable to perform the derived transformation.

42. (Original) The system of claim 34 wherein the source data schema is a source table schema describing source data tables, wherein the target data schema is a target table schema describing target data tables, and wherein the source table schema and the target table schema each describes at least one data table having columns.

43. (Original) The system of claim 42 wherein the source table schema is a source relational database schema describing source relational database tables, wherein the target table schema is a target relational database schema describing target database tables, and wherein the transformation is an SQL query.

44. (Previously Presented) The system of claim 43 wherein said mapping processor comprises:

a class identifier identifying at least one class in the ontology model corresponding to at least one table; and

a property identifier identifying at least one property or composition of properties in the ontology model corresponding to at least one table column.

45. (Previously Presented) The system of claim 44 wherein said property identifier presents a user with a choice of at least one property in the ontology model that may correspond to a given table column.

46. (Original) The system of claim 45 wherein the choice of at least one property only includes properties having targets that are compatible with a data type of the given table column.

47. (Original) The system of claim 46 wherein, for a given table column that is a foreign key to a foreign table, the choice of at least one property only includes properties whose target is a class corresponding to the foreign table.

48. (Previously Presented) The system of claim 43 wherein said transformation processor comprises:

an ontology labeler labeling properties of the ontology model with symbols;

a column converter converting at least one column in the source relational database schema into at least one source symbol, and converting at least one column in the target relational database schema into at least one target symbol; and

a symbol processor expressing the at least one target symbol in terms of at least one source symbol.

49. (Original) The system of claim 48 wherein said symbol processor uses expressions involving composition of properties.

50. (Original) The system of claim 48 wherein at least one dependency exists among properties in the ontology model, and wherein said transformation processor further comprises a dependency processor translating the at least one dependency among properties in the ontology model as at least one dependency between target relational database columns and source relational database columns, and wherein said symbol processor incorporates the at least one dependency between target relational database columns and source relational database columns.

51. (Original) The system of claim 50 wherein said symbol processor uses expressions involving arithmetic operations.

52. (Original) The system of claim 50 wherein said symbol processor uses expressions involving character string operations.

53. (Original) The system of claim 43 further comprising:

a data receiver receiving at least one source relational database table; and

a data processor applying the query to the at least one source relational database table to populate at least one target relational database table.

54. (Original) The system of claim 53 wherein the at least one source relational database table reside in a single database.

55. (Original) The system of claim 53 wherein the at least one source relational database table resides in multiple databases.

56. (Original) The system of claim 34 wherein the source data schema comprises a source document schema describing source documents, and wherein the target data schema comprises a target document schema describing target documents.

57. (Original) The system of claim 56 wherein the source document schema is a source DTD describing source XML documents, wherein the target document schema is a target DTD describing target XML documents, and wherein the source DTD and the target DTD each describes at least one XML element or XML attribute.

58. (Previously Presented) The system of claim 57 wherein the derived transformation is an XQuery.

59. (Previously Presented) The system of claim 57 wherein the derived transformation is an XSLT script.

60. (Original) The system of claim 56 wherein the source document schema comprises a source XML schema that describes XML source documents, wherein the target document schema comprises a target XML schema that describes XML target documents, and wherein the source XML schema and the target XML schema each comprises at least one XML complexType having at least one XML element or XML
attribute.

61. (Previously Presented) The system of claim 60 wherein the derived transformation is an XQuery.

62. (Previously Presented) The system of claim 60 wherein the derived transformation is an XSLT script.

63. (Previously Presented) The system of claim 60 wherein said mapping processor comprises:

a class identifier identifying at least one class in the ontology model corresponding to at least one XML complexType; and a property identifier identifying at least one property or composition of properties in the ontology model corresponding to at least one XML element or XML attribute.

64. (Original) The system of claim 60 wherein said transformation processor comprises an XML processor expressing XML elements and XML attributes of said target XML schema in terms of XML elements and XML attributes of said source XML schema.

65. (Original) The system of claim 64 wherein said XML processor operates recursively through XPath paths.

66. (Original) The system of claim 64 wherein at least one dependency exists among properties in the ontology model, and wherein said transformation processor further comprises a dependency processor translating the at least one dependency among properties in the ontology model as at least one dependency between target XML elements or attributes, and source XML elements or attributes, and wherein said XML processor incorporates the at least one dependency between target XML elements or attributes, and source XML elements or attributes.

67. (Previously Presented) The system of claim 60 further comprising:

a data receiver receiving at least one source XML document; and

a data processor applying the XSLT script to the at least one source XML document to generate at least one target XML document.

68. (Original) The system of claim 67 wherein the at least one source XML document reside in a single database.

69. (Original) The system of claim 67 wherein the at least one source XML document reside in multiple databases.

70-134. (Canceled)

135. (Currently Amended) An article of manufacture including one or more computer-readable media that embody a program of instructions for transforming data from one schema to another, wherein the program of instructions, when executed by a processing system, causes the processing system to:

receive a source data schema and a target data schema, the source data schema being different that the target data schema;

map the source data schema into an ontology model;

map the target data schema into the ontology model; and

derive a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target relational database schema, wherein the derived transformation ~~is adapted to~~ can transform the first data directly into the second data, and wherein the derived transformation ~~is adapted to~~ can transform the first data into the second data without transforming the first data via an ontological format.

136. (Original) The article of claim 135 wherein the one or more computer-readable media include one or more non-volatile storage devices.

137-140. (Canceled)